Mobile Sensing: Spring 2015
Exercise: 6

Due on 20th April 2015 by 17:45 PM.

Instructions: All course participants are requested to submit their exercise solutions electronically to the instructors (samuli.hemminki at cs.helsinki.fi and teemu.pulkkinen at cs.helsinki.fi), as well as to the course lecturer (petteri.nurmi at cs.helsinki.fi) by the due date (latest before the exercise session). In all the exercises, do not just give the answer, but also the derivation how you obtained it. Participants are encouraged to write computer programs to derive solutions to some of the given problems.

Ex 1. Load sensorData [1] including accelerometer and gyroscope measurements.

a) Write a function *rotate(G,gyro)* which rotates device inclination defined by gravity vector *G*, by given gyroscope measurements *gyro*. For the rotation, use rotation matrix: [1, -gyro(3), gyro(2); gyro(3), 1, -gyro(1); -gyro(2), gyro(1), 1], where ',' is used to separates columns and ';' is used to separates rows.

b) Write a function which implements complementary filtering based device inclination tracking (defined again by G): Incl(i) = (1-α)*accel(i) + α*Incl_{gyro}(i), where α=0.02, accel(i) is accelerometer measurements at index i, Incl_{gyro} = *rotate(Incl(i-1),gyro(i))*. Apply the function on the sensorData.

c) *Optional task* (Not counted for the mandatory 50%): Using the gravity estimation obtained in b), project the measuremens onto horizontal frame, and apply the PCA method described in lectures to find the direction of motion in device's local coordination system.

In the first exercise, subtasks a) and b) were probably most straight-forward to implement together. A small detail which caused some problems was that gyroscope measurements (which are in rad/sec) had to be scaled by a timestep dt. The last subtask c) was a relatively light task if you were already familiar with pca. Below is example code in Matlab:

[1] In the data, columns 1 to 7 corresponds to: [timestamp (in seconds), accelX, accelY, accelZ, gyroX, gyroY, gyroZ]

Example solutions for Exercise 6

```
%% Ex.1 a-c)

data = csvread('w6e1_sensorData.csv');

% Extract data
time  = data(:,1);
accel = data(:,2:4);
gyro  = data(:,5:7);

% Compute sampling frequency (timestamps in seconds)
sampleInterval = median(diff(time));
samplingFreq   = round(1/sampleInterval);

% PreAllocate
G           = zeros(size(accel));
gyroG       = zeros(size(accel));

% Initialize with first accelerometer measurement
frameNr     = size(accel,1);
initG       = accel(1,:);
gyroG(1,:) = initG;
G(1,:)      = initG;

% Weight parameter for complementary filterting
alpha = 0.02;

% Loop through data
for i=2:frameNr

    % Stepsize for rotation
    gyro(i,:) = gyro(i,:) .* sampleInterval;

    % Rotation Matrix
    rotMatrix = [1 -gyro(i,3) gyro(i,2); ...
                  gyro(i,3) 1 -gyro(i,1); ...
                 -gyro(i,2) gyro(i,1) 1];

    gyroG(i,:) = G(i-1,:) * rotMatrix;
    G(i,:) = (1-alpha)*gyroG(i,:) + alpha*accel(i-1,:);

end

% Project onto vertical/horizontal planes
L = zeros(frameNr,3);
V = zeros(frameNr,3);
H = zeros(frameNr,3);

for i=1:length(time)
```

```
    % Remove gravity: Data - G
    L(i,:) = accel(i,:) - G(i,:);

    % Project data to direction of gravity (vertical)
    V(i,:) = (dot(L(i,:),G(i,:))/dot(G(i,:),G(i,:))) * G(i,:);

    % Horizontal Acc. = Total Lin.Acc. - Vertical Acc.
    H(i,:) = L(i,:)-V(i,:);

end

% Perform PCA
p = pca(H);

% unit vector of the first principal component
p1 = p(:,1) / norm(abs(p1));
```

Ex 2. Load magnetometer measurements [2] collected using a smartphone during a brief walk inside Exactum 3rd floor.

a) Write a function *Compass(m)* (m contains magnetometer measurements), that calculates angle between device's Y-axis and magnetic north. You can assume that the device is in neutral orientation, where pitch and roll are equal to zero (i.e., no tilt compensation for magnetometer required) and a standard sensor coordination system.

b) Write a function *QuaternionsToEulers(qtrn)*, which translates quaternion orientation into euler angles roll, pitch and yaw. Translate quaternion orientation collected during walking to euler angles; see lecture VI slides for details.

In the Ex.2 a), the compass reading, i.e., angle from north, could be simply use the formulas from the lectures. As a comparison, below is also quaternions conversion using rotation matrix in a common ZYX format. Below is example code in Matlab:

```
%% Ex. 2 a)

clear;
data = csvread('w6e2_magnetometer.csv');
time = data(:,1);
magn = data(:,2:4);

% Compass
heading = zeros(length(time),1);
x = magn(:,1);
y = magn(:,2);
```

[2] In the data, columns 1 to 4 corresponds to: [timestamp (in seconds), magnX, magnY, magnZ]

```
for i=1:length(time)

    a = rad2deg(atan(y(i)/x(i)));

    if x(i) < 0
        heading(i) = 180 - a;
    elseif x(i) > 0 && y(i) < 0
        heading(i) = - a;
    elseif x(i) > 0 && y(i) > 0
        heading(i) = 360 - a;
    elseif x(i) == 0 && y(i) < 0
        heading(i) = 90;
    elseif x(i) == 0 && y(i) > 0
        heading(i) = 270;
    end

end

%% Ex. 2 b)

clear;
q = csvread('w6e2_quaternions.csv');

% Quaternions to ZYX eulers (Rotation matrix method)

% First convert to rotation matrix (only a subsection of RM required)
R(1,1,:) = 2.*q(:,1).^2-1+2.*q(:,2).^2;
R(2,1,:) = 2.*(q(:,2).*q(:,3)-q(:,1).*q(:,4));
R(3,1,:) = 2.*(q(:,2).*q(:,4)+q(:,1).*q(:,3));
R(3,2,:) = 2.*(q(:,3).*q(:,4)-q(:,1).*q(:,2));
R(3,3,:) = 2.*q(:,1).^2-1+2.*q(:,4).^2;

% Then rotation matrix to eulers is easy:
phi = atan2(R(3,2,:), R(3,3,:) );
theta = -atan(R(3,1,:) ./ sqrt(1-R(3,1,:).^2) );
psi = atan2(R(2,1,:), R(1,1,:) );

euler = [phi(1,:)' theta(1,:)' psi(1,:)']';

% Quaternions to XYZ eulers (Lecture slides)
psiL   = zeros(length(q),1);
thetaL = zeros(length(q),1);
phiL   = zeros(length(q),1);

for i=1:length(q)
    psiL(i)   = atan2( 2*(q(i,1)*q(i,2)+q(i,3)*q(4)), 1-2*(q(i,2)^2+q(i,3)^2) );
    thetaL(i) = asin( 2*(q(i,1)*q(i,2)-q(i,4)*q(i,2)) );
    phiL(i)   = atan2( 2*(q(i,1)*q(i,4)+q(i,2)*q(i,3)), 1-2*(q(i,3)^2+q(i,4)^2) );
end
```

```
eulerL = [phiL real(thetaL) psiL]';
```

Ex 3. Besides providing information about device's orientation, measurements about magnetic fields can be also used for fingerprinting based indoor localization. Read the article Indoor Location Sensing Using Geo-Magnetism by J. Chung et al.

a) Summarize, using up to half a page, the initial system and evaluation. (Section 3)

b) Summarize, using up to half a page, the pedestrian localization setup and results (Section 4)